CLAIMS

1. A method for providing a remembered set comprising:

maintaining the remembered set as a bag;

identifying when an event occurs; and

transforming the remembered set into a set when the event occurs.

2. The method of claim 1 wherein the step of transforming further comprises:

obtaining a plurality of thread local store buffers; and

flushing the thread local store buffers to a global store buffer.

3. The method of claim 2 wherein the step of flushing further comprises:

eliminating a plurality of objects in the global store buffer that reside in the thread local store buffers wherein the objects are duplicates.

4. The method of claim 3 wherein the step of eliminating further comprises:

determining whether one or more dedicated mark bits are set in the objects referenced by the thread local store buffers; and

pushing the objects to the global store buffer, if said mark bits are not set.

marking a plurality of the dedicated mark bits, if they are not set.

5. The method of claim 2 wherein the event occurs when the thread local store buffers have overflowed.

6. The method of claim 5 wherein the step of identifying further comprises:

using a memory protection mechanism to determine when the thread local store buffers has overflowed.

7. The method of claim 5 wherein the step of identifying further comprises:

using a user trap to determine when the thread local store buffers has overflowed.

8. The method of claim 7 wherein the user trap comprises:

filling a stack with a plurality of pointers to successive stack elements;

filling a top of the stack with a pointer to the user trap;

fetching a current pointer from the pointers to successive stack elements;

determining whether the current pointer is the pointer to the user trap; and

performing the user trap, if the current pointer is the pointer to the user trap.

9. The method of claim 2 wherein the step of flushing further comprises:

allocating a chunk in the global store buffer wherein the chunk is the same size as one of the thread local store buffers.

10. The method of claim 3 wherein the step of flushing further comprises:

allocating a chunk in the global store buffer wherein the chunk is smaller than a current buffer in the thread local store buffers; and

shrinking the current buffer to the size of the chunk.

11. The method of claim 10 further comprising:

determining a variable threshold.

12. The method of claim 11 further comprising:

failing to shrink the current buffer to the size of the chunk if the variable threshold is larger than the chunk.

13. A remembered set comprising:

a first remembered set configured to be maintained as a bag;

an event configured to be identified when it occurs; and

a second remembered set configured to be transformed from the first remembered set when the event occurs wherein the second remembered set is a set.

14. The remembered set of claim 13 wherein the second remembered set comprises:

a flushing mechanism configured to flush a plurality of thread local store buffers into the global store buffer.

15. The remembered set of claim 14 wherein the flushing mechanism further comprises:

a plurality of objects that reside in the thread local store buffers configured to be eliminated from the global store buffer wherein the objects are duplicates.

16. The remembered set of claim 15 wherein the flushing mechanism further comprises:

one or more dedicated mark bits in the objects referenced by the thread local store buffers wherein the dedicated mark bits are configured to be determined if they are set and wherein the objects are configured to be pushed onto the global store buffer, if said mark bits are not set.

17. The remembered set of claim 14 wherein the event occurs when the thread local store buffers have overflowed.

18. The remembered set of claim 17 wherein the event further comprises:

a memory protection mechanism configured to determine when the thread local store buffers has overflowed.

19. The remembered set of claim 17 wherein the event further comprises:

a user trap configured to determine when the thread local store buffers has overflowed.

20. The remembered set of claim 19 wherein the user trap comprises:

a stack configured to be filled with a plurality of pointers to successive stack elements;

a top of the stack configured to be filled with a pointer to the user trap;

a current pointer configured to be fetched from the pointers to successive stack elements wherein it is determined whether the current pointer is the pointer to the user trap; and

the user trap, configured to be performed if the current pointer is the pointer to the user trap.

21. The remembered set of claim 14 wherein the flushing mechanism further comprises:

a chunk in the global store buffer configured to be allocated wherein the chunk is the same size as one of the thread local store buffers.

22. The remembered set of claim 15 wherein the flushing mechanism further comprises:

a chunk in the global store buffer configured to be allocated wherein the chunk is smaller than a current buffer in the thread local store buffers and wherein the current buffer is configured to be shrunk to the size of the chunk.

23. The remembered set of claim 22 further comprising:

a variable threshold configured to be determined.

24. The remembered set of claim 23 further comprising:

the current buffer configured not to be shrunk to the size of the chunk if the variable threshold is larger than the chunk.

25. A computer program product comprising:

a computer usable medium having computer readable program code embodied therein configured to provide a remembered set, the computer program product comprising:

computer readable code configured to cause a computer to maintain the remembered set as a bag;

computer readable code configured to cause a computer to identify when an event occurs; and

computer readable code configured to cause a computer to transform the remembered set into a set when the event occurs.

26. The computer program product of claim 25 wherein the computer readable code configured to cause a computer to transform further comprises:

computer readable code configured to cause a computer to obtain a plurality of thread local store buffers; and

computer readable code configured to cause a computer to flush the thread local store buffers to a global store buffer.

27. The computer program product of claim 26 wherein the computer readable code configured to cause a computer to flush further comprises:

computer readable code configured to cause a computer to eliminate a plurality of objects in the global store buffer that reside in the thread local store buffers wherein the objects are duplicates.

28. The computer program product of claim 27 wherein the computer readable code configured to cause a computer to eliminate further comprises:

computer readable code configured to cause a computer to determine whether one or more dedicated mark bits are set in the objects referenced by the thread local store buffers; and

computer readable code configured to cause a computer to mark a plurality of the dedicated mark bits, if they are not set.

29. The computer program product of claim 26 wherein the event occurs when the thread local store buffers have overflowed.

30. The computer program product of claim 29 wherein the computer readable code configured to cause a computer to identify further comprises:

computer readable code configured to cause a computer to use a memory protection mechanism to determine when the thread local store buffers has overflowed.

31. The computer program product of claim 29 wherein the computer readable code configured to cause a computer to identify further comprises:

computer readable code configured to cause a computer to use a user trap to determine when the thread local store buffers has overflowed.

32. The computer program product of claim 31 wherein the user trap comprises:

computer readable code configured to cause a computer to fill a stack with a plurality of pointers to successive stack elements;

computer readable code configured to cause a computer to fill a top of the stack with a pointer to the user trap;

computer readable code configured to cause a computer to fetch a current pointer from the pointers to successive stack elements;

computer readable code configured to cause a computer to determine whether the current pointer is the pointer to the user trap; and

computer readable code configured to cause a computer to perform the user trap, if the current pointer is the pointer to the user trap.

33. The computer program product of claim 26 wherein the computer readable code configured to cause a computer to flush further comprises:

computer readable code configured to cause a computer to allocate a chunk in the global store buffer wherein the chunk is the same size as one of the thread local store buffers.

34. The computer program product of claim 27 wherein the computer readable code configured to cause a computer to flush further comprises:

computer readable code configured to cause a computer to allocate a chunk in the global store buffer wherein the chunk is smaller than a current buffer in the thread local store buffers; and

computer readable code configured to cause a computer to shrink the current buffer to the size of the chunk.

35. The computer program product of claim 34 further comprising:

computer readable code configured to cause a computer to determine a variable threshold.

36. The computer program product of claim 35 further comprising:

computer readable code configured to cause a computer to fail to shrink the current buffer to the size of the chunk if the variable threshold is larger than the chunk.